

Table of Contents

Requirements	1
System requirements	1
Required plugins	1
1. Step: Setting up Codabix	1
1.1 Installation	2
1.2 Configuration	2
2. Step: Setting up the process data acquisition	3
2.1 Creating a new transfer	4
2.2 Selection of the data to be recorded	6
2.3 Selection of data targets	9
2.4 Defining the trigger	10
3. Step: Activate and verify functionality	11
3.1 View in the SQLite browser	12
3.2 View in the Codabix dashboard	14

Capturing process data

These instructions describe the “step by step” procedure to write process data based on definable triggers into a database.

In tutorial a **SIMATIC S7 PLC** is used as data source and a local **SQLite Database** as data target. As Codabix offers a uniform interface that is used to access connected devices and databases, this procedure is applicable to **any type of data**, that is defined in Codabix.

Requirements

System requirements

To carry out the following steps you need a computer (running Windows or Linux), on which you have the necessary rights to install applications.

Furthermore, an internet connection is necessary to load Codabix and the provided standard configuration and to be able to access sample data from our publicly available SIMATIC S7 PLC.

The exact system requirements for hardware and operating system version can be found here:

- [System requirements for Windows](#)
- [System requirements for Linux](#)

Required plugins

The following Codabix plugins are required

- [S7 Device Plugin](#): Connection to the SIMATIC S7 PLC
- [Script Interface Plugin](#): Processing the triggers and transferring the data

1. Step: Setting up Codabix


In this step you install Codabix on your system.

For the purpose of this tutorial we provide you with a basic configuration for the process data acquisition.

1.1 Installation

- Download in the [Download area](#) the current version of Codabix (these instructions require **at least v1.4.0**) for the system, where you want to run Codabix.
- After a successful download, install Codabix on your system and follow the steps for the first start:
 - [Windows](#)
 - [Linux and Raspberry Pi](#)
 - [Using the Siemens IOT2050 Image](#)

1.2 Configuration

- Download the following default configuration: [capture-process-data_default_2023-07-28.cbx](#)
- Import the configuration into Codabix
 - For **Codabix v1.0.0 and above** (including the preview versions) it is now possible to restore a backup via the web configuration:
 - Open the Codabix web configuration in a browser as described [here](#)
 - Click on the sidebar menu entry **Backup & Restore**
 - In the section **Restore Backup** click on the button **Select Backup file...**
 - In the **Open File** dialog: Select the default configuration file you downloaded in the previous step
 - After uploading the file to the server click the button **Restore Now...** to restore the backup
 - Alternatively, you can restore the backup in the Codabix application:
 - **Windows**
 - Click  on the right side of the taskbar
 - In the backup dialog, select the previously loaded file and start the import by clicking on **Restore Now...**

Restore Backup

When restoring a backup, the project directory and the back-end database will be overwritten with the contents from the backup.

Backup file:
C:\Users\fabian\Downloads\capture-process-data_default.cbx

Backup Project Name:	Industrie 4.0 Use Case: Erfassung von Prozessdaten
Backup Date:	12/18/2019 15:05:20
Backup CoDaBix Version:	0.21.0
Includes Password Key:	Yes
Includes History Values:	No

Project Directory: C:\Users\fabian\workspace\test\my-codabix-data Change...

☒ Keep Current Project Settings Edit...

☐ Restore Project Settings from Backup Edit...

☐ Restore Log Files

Restore Now... Cancel

- **Linux/Raspberry Pi/Siemens IOT2050**

- Select the menu item 4) Restore Backup in the console application
- Enter the path to the loaded file and start the import by pressing the Enter-key

- **Docker**

- The procedure for restoring a backup within a Docker container can be found in the associated Github repository:
<https://github.com/Traeger-GmbH/codabix-docker#restoring-from-a-backup-file>

2. Step: Setting up the process data acquisition

In this example, the values of two variables (Duration and Quantity) will be read from the PLC and a new entry will be created in a database table, which also contains the MachineID and the current timestamp. The MachineID can be used to associate the database entry with the source PLC. This process should be carried out when the rising edge of the JobCompleted bit is detected in the PLC.

The **PLC variables** are defined as follows:

Process data

- Duration
 - Datatype: Real
 - Address: DB511.DBD 0
- Quantity
 - Datatype: DInt
 - Address: DB511.DBD 4

Trigger

- JobCompleted
 - Datatype: Bit
 - Address: DB511.DBX 10.0

The **database table** named oeemachinerecord has the following columns:

- ID
 - Datatype: INT
- MachineID
 - Datatype: INT
- Duration
 - Datatype: Double
- Quantity
 - Datatype: INT
- Timestamp
 - Datatype: DATETIME

In order to map the data acquisition (henceforth referred to as Transfer) in Codabix, it is divided into three components:

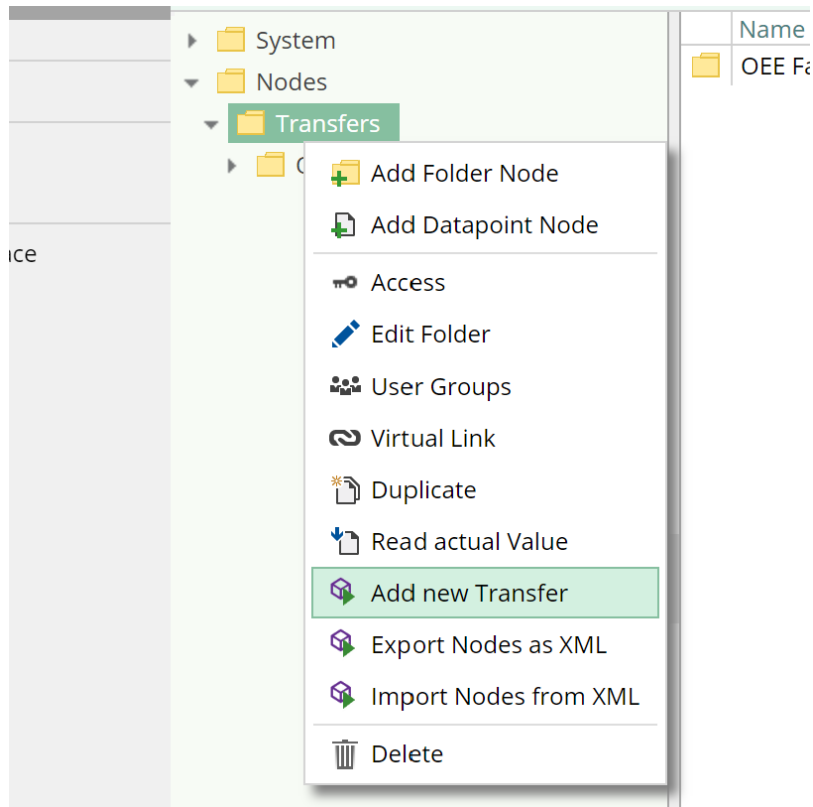
- **Inputs:** Variables whose values are to be recorded
- **Outputs:** Variables in which the data to be recorded will be saved
- **Triggers:** Variables that trigger a data transfer from the **Inputs** to the **Outputs**

These components are mapped using a folder structure within Codabix. Each folder that is located under the path /Nodes/Transfers represents an independent transfer.

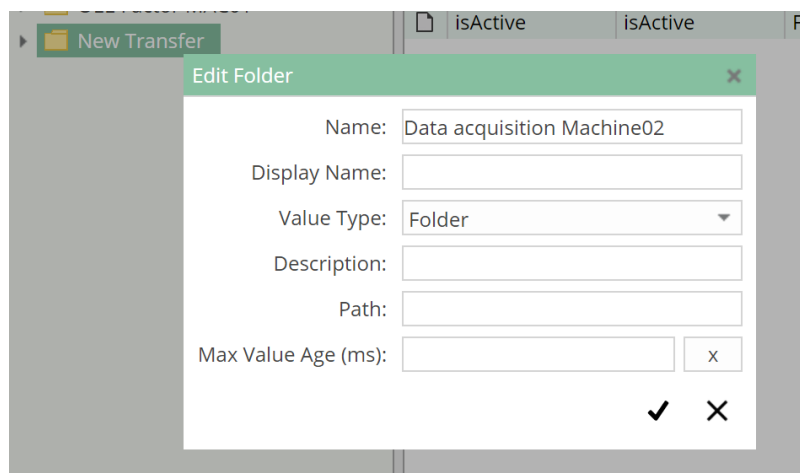
2.1 Creating a new transfer

To create a new transfer, navigate to the /Nodes/Transfers node in the node tree.

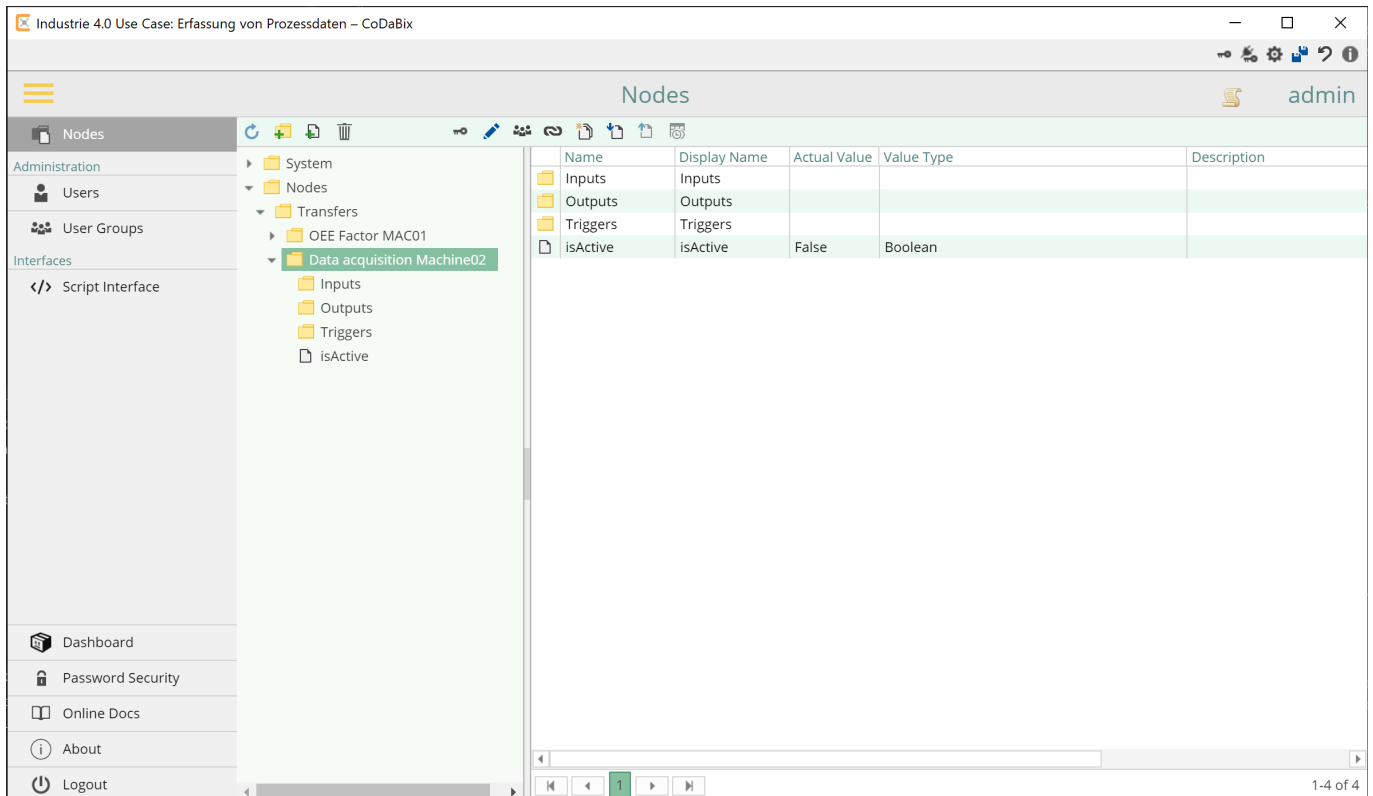
- Create a new transfer:
 - Right click on the node /Nodes/Transfers
 - In the context menu click on the entry Add new Transfer



- Name the new transfer:
 - Right click on the new node New Transfer
 - Click on the context menu entry Edit Folder
 - In the Edit Folder dialog, change the name to Data acquisition Machine02
 - Confirm by clicking on



The node tree should then look like this:



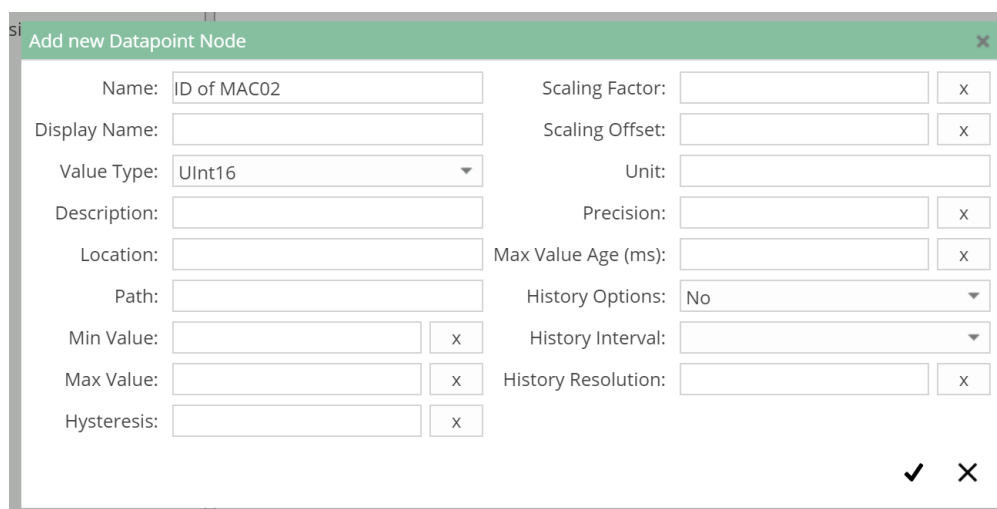
Name	Display Name	Actual Value	Value Type	Description
Inputs	Inputs			
Outputs	Outputs			
Triggers	Triggers			
isActive	isActive	False	Boolean	

2.2 Selection of the data to be recorded

In order to define variables whose values are to be recorded, these variables must be located under the folder node Inputs. In addition to the variables from the PLC, the ID of the machine (for this example MachineID = 2) shall be recorded.

Therefore, first a node of the type UInt16 is created that will hold this MachineID as constant value:

- Right click on the node Inputs
- Select Add Datapoint Node from the context menu
- In the Add Datapoint Node dialog:
 - Enter ID of MAC02 as the name
 - Select UInt16 as **Value Type**
 - Confirm by clicking on ✓

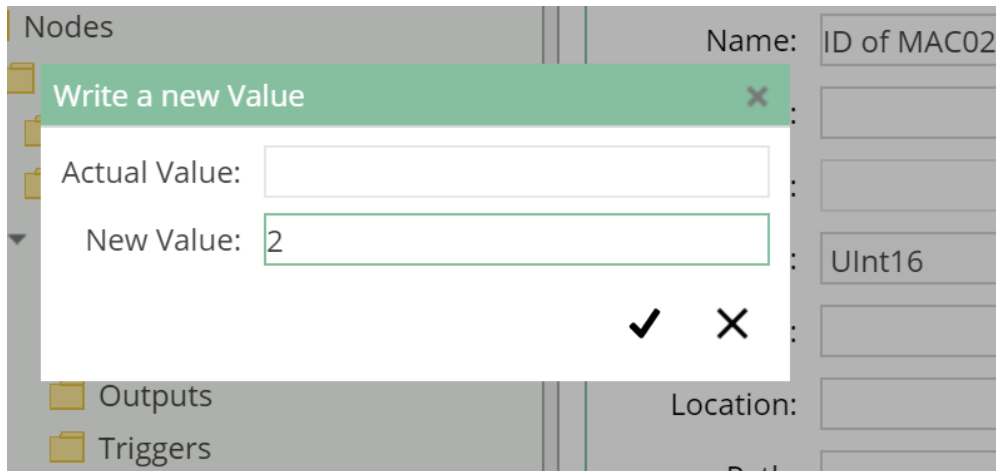


Add new Datapoint Node

Name:	ID of MAC02	Scaling Factor:		x
Display Name:		Scaling Offset:		x
Value Type:	UInt16	Unit:		
Description:		Precision:		x
Location:		Max Value Age (ms):		x
Path:		History Options:	No	
Min Value:		History Interval:		
Max Value:		History Resolution:		x
Hysteresis:				x

✓ ✕

- Right click on the new node
- Select Write a new Value
- In the Write a new Value dialog:
 - Enter the value 2 as the new value (as this Transfer will record the data from the machine with MachineID = 2)
 - Confirm by clicking on ✓



To define the values of the PLC variables as inputs for the transfer, a linked node is created in the Inputs folder whose link target contains the value node that represents the corresponding PLC variable.

Linked nodes behave similarly to symbolic links as they are available in the Unix file system. They offer the advantage, that the structure of data can be designed independently without influencing the original structure.

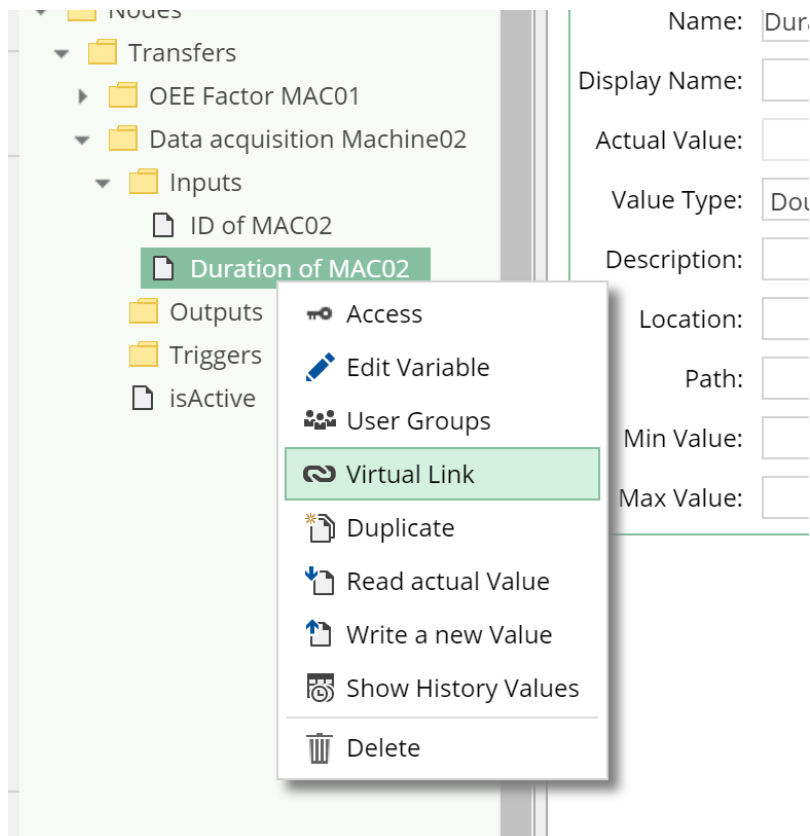
The nodes that represent these variables can be found under the path /System/Devices/S7 TCP-IP Device/Channels/Machine 02/Variables:

The screenshot shows the CoDaBix interface. On the left, the 'Nodes' tree is expanded to show the path: System > Devices > S7 TCP-IP Device > Channels > Machine02 > Variables. The 'Variables' folder is highlighted. On the right, a table displays the variables and their current values.

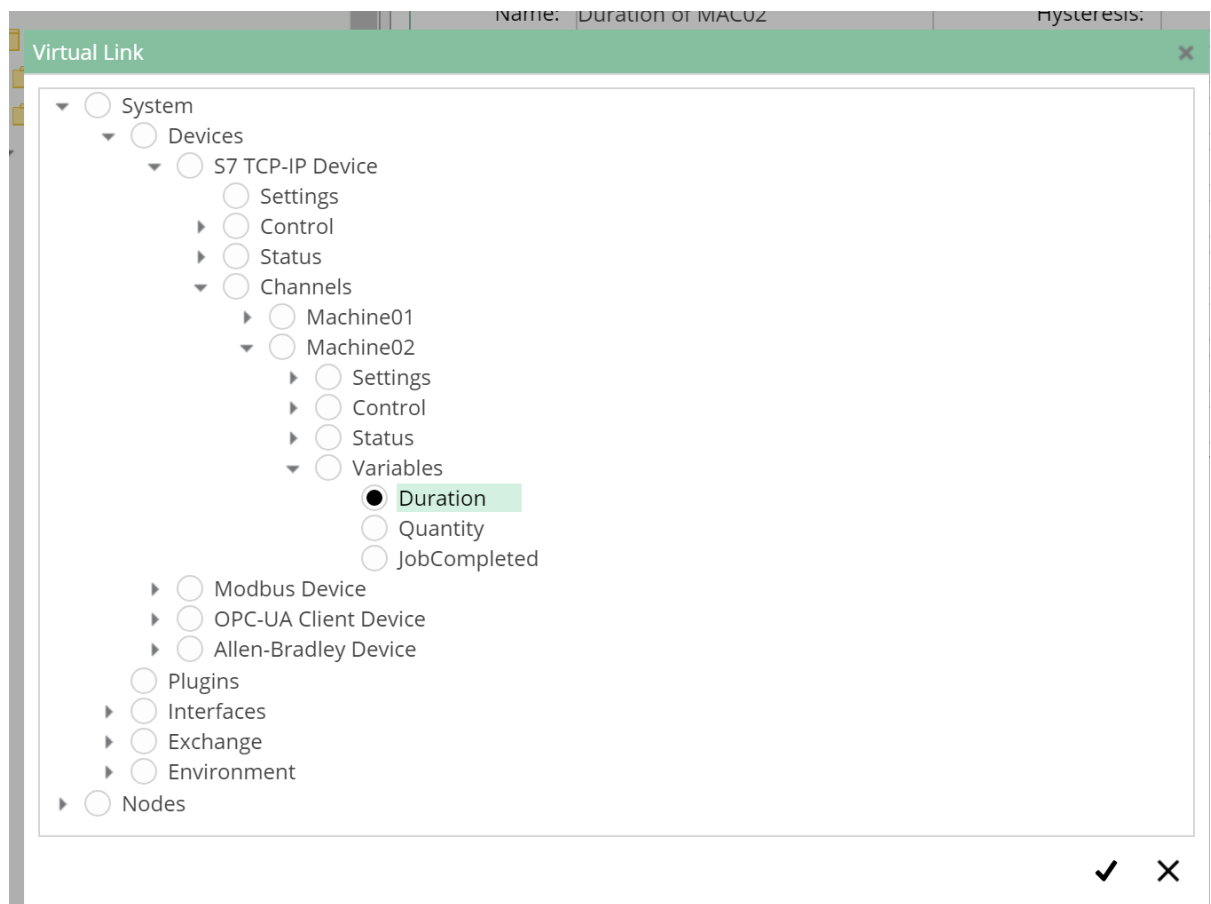
Name	Display Name	Actual Value
Duration	Duration	9925
Quantity	Quantity	3
JobCompleted	JobCompleted	False

For these two nodes, do the following to link them in the Inputs folder:

- Right click on Inputs
- Click on the context menu Add Datapoint Node
- In the Add Datapoint Node dialog:
 - Enter the name of the original node as name (the name of the link node can be freely selected, but in this case it should be obvious which node is the link target)
 - Confirm by clicking on ✓
- Right click on the newly created node
- In the context menu, select the option Virtual Link



- In the Virtual Link dialog:
 - In the opened dialog, navigate to the node to which the virtual link should point
 - Confirm by clicking on ✓



Since the current timestamp shall be saved in addition to the PLC data, a node must be added that delivers this value.

For this purpose Codabix provides a predefined value node, which can be found under the path `/System/Environment/DateTime/UtcNow`. When reading this node always the current timestamp is returned.

Now, in the same way as in the previous steps, create another link called `Current Timestamp` that refers to this node (`/System/Environment/DateTime/UtcNow`).

2.3 Selection of data targets

Analogously to the sources of the data, the destinations are defined under the folder node `Outputs`.

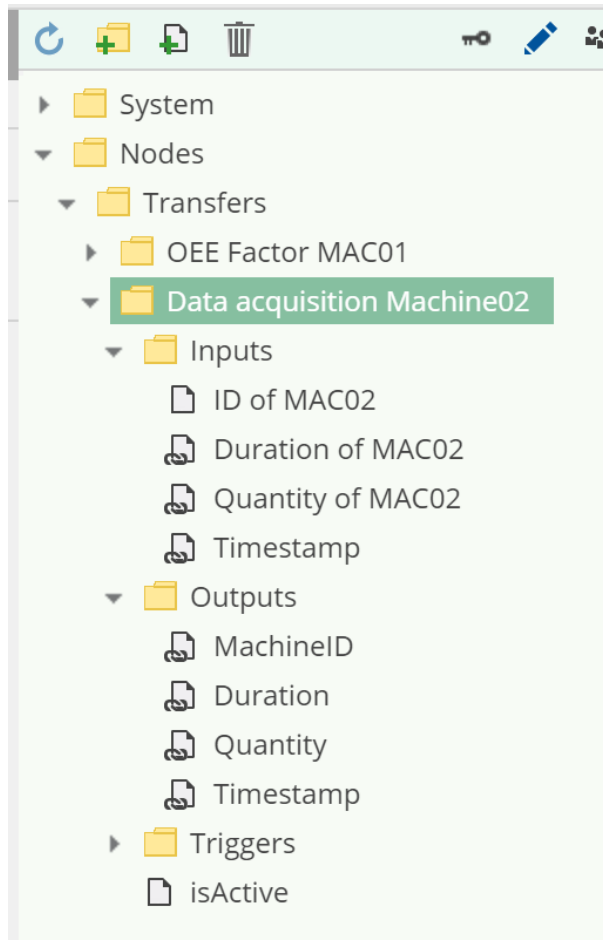
The table `oemmachinerecord` of the database will be used for this. The nodes that represent the columns of this table are under the path `/System/Exchange/SQL`

`Exchange/Databases/OEE/Tables/oemmachinerecord/Columns`. As in the previous step, create link nodes below `Outputs` that are linked to the following nodes:

- **MachineID:** `/System/Exchange/SQL
Exchange/Databases/OEE/Tables/oemmachinerecord/Columns/MachineID`
- **Duration:** `/System/Exchange/SQL
Exchange/Databases/OEE/Tables/oemmachinerecord/Columns/Duration`
- **Quantity:** `/System/Exchange/SQL
Exchange/Databases/OEE/Tables/oemmachinerecord/Columns/Quantity`
- **Timestamp:** `/System/Exchange/SQL`

Exchange/Databases/OEE/Tables/oemachinerecord/Columns/Timestamp

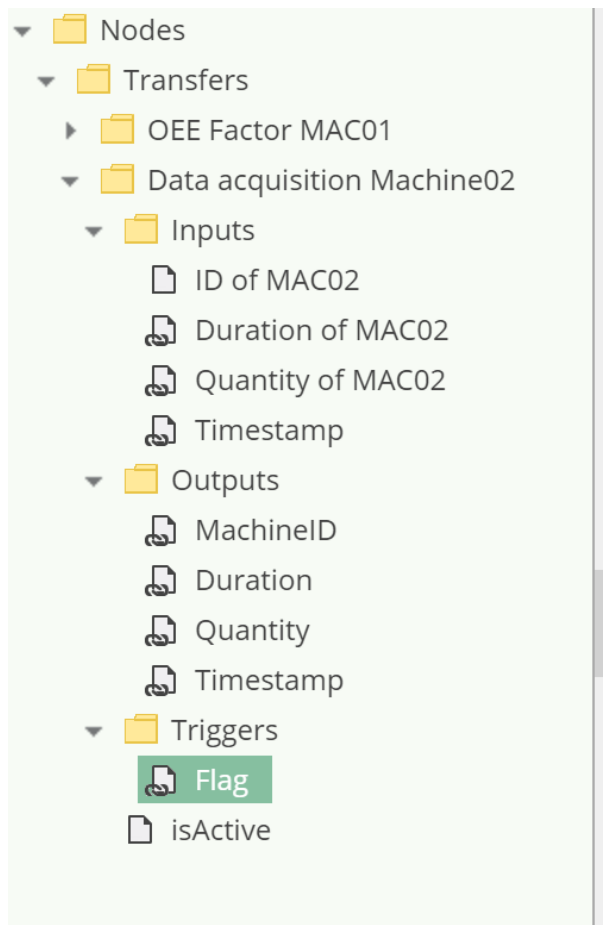
The transfer mechanism uses the order of the input and output nodes to differentiate which data should be written to which node. The value of the first input node is written to the first output node, the value of the second to the second output node and so on. You can use drag-and-drop to arrange the nodes under Inputs and Outputs so that their order matches the order in this screenshot:



2.4 Defining the trigger

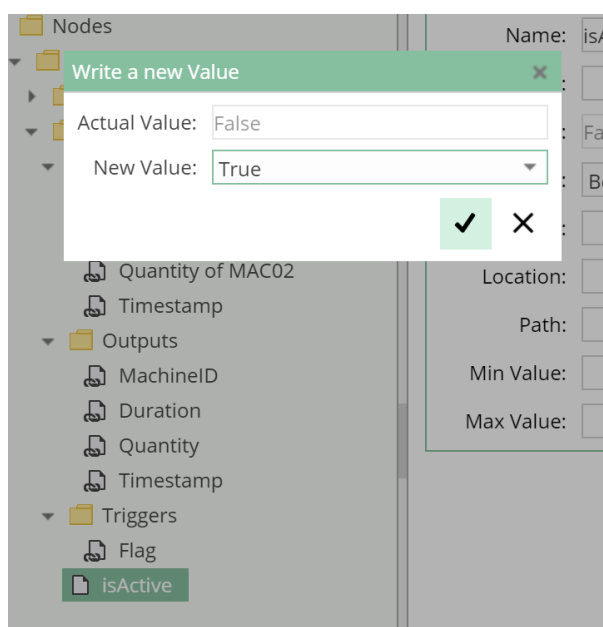
To set up the trigger for our example, create a virtual link under the Triggers folders, whose target is the node /System/Devices/S7 TCP-IP Device/Channels/Machine 01/Variables/JobCompleted.

So the node tree will look like this:



3. Step: Activate and verify functionality

After completing the previous steps, the data acquisition setup is complete. To activate the transfer, the value of the node `isActive` in the folder of the transfer must be set to `true`. To do this, write `true` as the new node value:



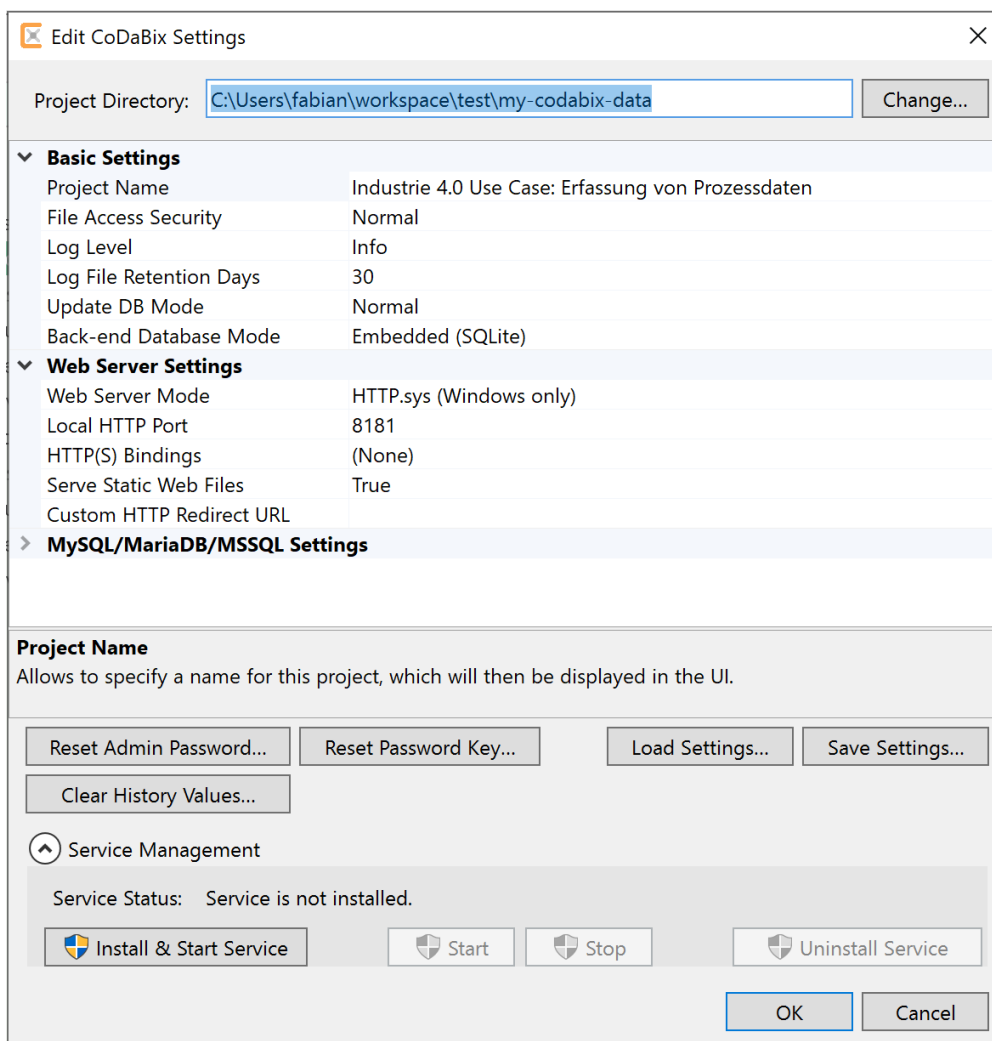
The trigger is now read by Codabix every 500ms and checked for a rising edge. If this is the case, the inputs are read, then the trigger is reset (the value `false` is written) and the data is written to the outputs. This will create a new entry in the database table that contains this data.

The PLC program waits until the trigger has been reset. After that within 10 seconds new, random values are written to the variables and the trigger is set on `true` again.

3.1 View in the SQLite browser

You can now verify the functionality using an SQLite browser:

- If you have not already installed an SQLite browser on your system, download the .zip archive that is suitable for your system at <https://sqlitebrowser.org/dl/>
- Unzip and start the SQLite browser
- Open the database file under the following path `<CodabixProjectDirectory>/userdata/mydata.sqlite`.
`<CodabixProjectDirectory>` stands for the path of the project directory on your operating systems filesystem, that you select when you started Codabix for the first time. This path can be found in the settings dialog:



- Open the table `oeemachinerecord` in the SQLite browser

The screenshot shows the 'DB Browser for SQLite' application window. The title bar indicates the file path: 'C:\Users\fabian\workspace\test\my-codabix-data\userdata\mydata.sqlite'. The menu bar includes 'File', 'Edit', 'View', 'Tools', and 'Help'. The toolbar contains buttons for 'New Database', 'Open Database', 'Write Changes', 'Revert Changes', 'Open Project', 'Save Project', 'Attach Database', and 'Close Database'. The 'Database Structure' tab is active, displaying a tree view of the database schema. The 'oeemachine' table is selected, showing two records in a table view:

ID	Machine
1	Machine01
2	Machine02

The 'Edit Database Cell' panel on the right is open, showing the 'Text' mode for the selected cell. The 'Remote' panel at the bottom right is also visible, showing a table with columns 'Name', 'Commit', 'Last modified', and 'Size'. The status bar at the bottom indicates 'UTF-8' encoding.

- You should now find existing entries there, while new ones are added every 10 seconds

DB Browser for SQLite - C:\Users\fabian\workspace\test\my-codabix-data\userdata\mydata.sqlite

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: oeemachinerecord

	ID	MachineID	Duration	Quantity	Timestamp
	Filter	Filter	Filter	Filter	Filter
1	527	2	7437.0	10	2019-12-18 15:10:41....
2	528	2	6563.0	1	2019-12-18 15:10:44....
3	529	2	7644.0	1	2019-12-18 15:10:50....
4	530	2	5375.0	7	2019-12-18 15:10:50....
5	531	2	6014.0	8	2019-12-18 15:11:09....

New Record Delete Record

Edit Database Cell

Mode: Text Import Export Set as NULL

527

Type of data currently in cell: Text / Numeric
3 char(s) Apply

Remote

Identity

Name	Commit	Last modified	Size
------	--------	---------------	------

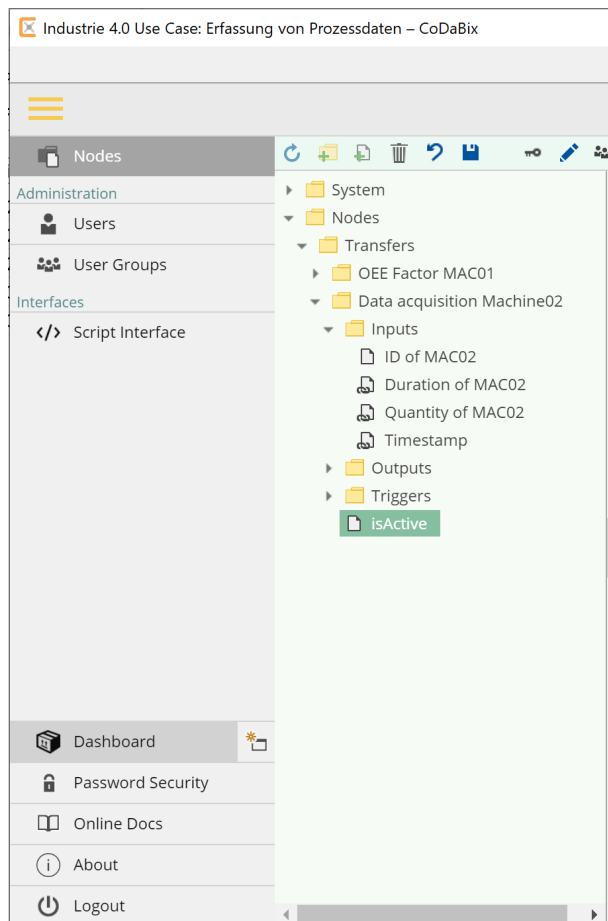
1 - 5 of 5 Go to: 1

SQL Log Plot DB Schema Remote UTF-8

3.2 View in the Codabix dashboard

Another possibility is the view in the dashboard application integrated in Codabix.

- To do this, click on the menu item Dashboard in the left sidebar



- Navigate to the folder Data acquisition Machine02
- Expand the three folders Inputs, Outputs and Triggers by clicking on the ▼ symbols next to them
- You can now see the change in the node values in almost real time and can thus observe the triggering and resetting of the trigger and the transfer of data between the inputs and outputs live:

Nodes

Administration

Users

User Groups

Interfaces

</>

Script Interface

Dashboard

Password Security

Online Docs

About

Logout

Dashboard

admin

>

Nodes

>

Transfers

>

Data acquisition Machine02

>

Inputs

ID of MAC02

2

Duration of MAC02

5305

Quantity of MAC02

10

Timestamp

12/18/2019, 4:14 PM

>

Outputs

MachineID

2

Duration

5305

Quantity

10

Timestamp

12/18/2019, 4:14 PM

>

Triggers

Flag

isActive

Version: 0.0.28

From:

<https://codabix.de/> - **CoDaBix®**

Permanent link:

<https://codabix.de/en/tutorials/capturing-process-data>

Last update: **2023/07/28 11:35**